Data Understanding and Preparation

Rita P. Ribeiro

Machine Learning - 2021/2022

CRISP-DM: a Typical Data Mining Workflow

- Cross-Industry Process for Data Mining (CRISP-DM)

Shearer C.: The CRISP-DM model: the new blueprint for data mining, J Data Warehousing (2000); 5:13—22.

CRISP-DM: Data Understanding

- Collect Initial Data:
  initial data collection report
- Describe Data:
  data description report
- Explore Data:
  data exploration report
- Verify Data Quality:
  data quality report

CRISP-DM: Data Preparation

- Data Set:
  data set description
- Select Data:
  rationale for inclusion/exclusion
- Clean Data:
  data cleaning report
- Construct Data:
  derived variables, generated records
- Integrate Data:
  merged data
- Format Data:
  reformatted data

Summary

- Data Understanding
  - Data Quality
  - Data Summarization
  - Data Visualization
- Data Preparation
  - Feature Extraction
  - Data Cleaining
  - Data Transformation
  - Feature Engineering
  - Data and Dimensionality Reduction

Data Understanding

Data Summarization

- Motivation
  - With big data sets it is hard to have an idea of what is going on in the data
  - Data summaries provide overviews of key properties of the data
  - Help selecting the most suitable tool for the analysis
  - Their goal is to describe important properties of the distribution of the values
- Types of Summaries
  - What is the “most common value”?
  - What is the “variability” in the values?
  - Are there “strange” / unexpected values in the data set?

Data Summarization (cont.)

- Data set
  - Univariate data
  - Multivariate data
- Variables
  - Categorical variables
  - Numeric variables

Data Summarization (cont.)

Example Data set

- `algae` data set composed by 200 water samples taken at several European rivers, which are described by:
  - 3 categorical variables: `season`, `size` and `speed` of the river
  - 8 numeric variables with chemical concentration measurements
  - 7 numeric variables with the concentration level of harmful algae.

Data Summarization: Categorical Variables

- Mode: the most frequent value
- Frequency table: frequency of each value (absolute or relative)
  - `season`

autumn	spring	summer	winter
40	53	45	62

- Contingency tables: cross-frequency of values for two variables
  - `season` and `size`

	autumn	spring	summer	winter
large	11	12	10	12
medium	16	21	21	26
small	13	20	14	24

Data Summarization: Numeric Variables

Statistics of location

- Mean (or sample mean) - sensitive to extreme values

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Median
  - It is the 50^{th}-precentile, i.e. the value above (below) which there are 50% of the values in the data set
- Mode
  - It is the most common (more frequently occurring) value in a set of values
    - Note that the mode can be applied to categorical variables

Data Summarization: Numeric Variables (cont.)

Statistics of variability or dispersion

- Range: $max_x - min_x$
- Variance σ_x^2 - sensitive to extreme values
- Standard Deviation - sensitive to extreme values

$$\sigma_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)^2}$$

- Inter-quartile Range (*IQR*)
  - It is the difference between the 3rd (Q_3) and 1st (Q_1) quartiles
    - Q_1 is the number below which there are 25% of the values
    - Q_3 is the number below which there are 75% of the values

Data Summarization: Numeric Variables (cont.)

"An outlier is a point that deviates so much from the other data points as to arouse suspicions that it was generated by a different mechanism" (Hawkins, 1980)

- For a numeric variable an outlier can be an extreme value
- In the presence of such values,
  - median or mode are more robust as a central tendency statistic
  - inter-quartile range is more appropriate as variability statistic.
- Boxplot definition (Tukey, 1977)
  - any value outside the interval $[Q_1 - 1.5 \times IQR, Q_3 + 1.5 \times IQR]$ is an outlier

Multivariate analysis of variability or dispersion

- Covariance Matrix: variance between every pair of numeric variables - the value depends on the magnitude of the variable

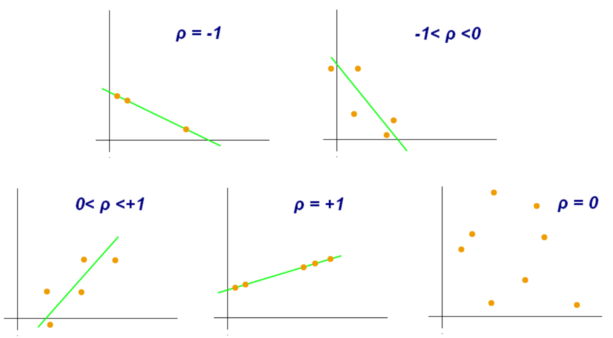
$$cov(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)$$

- Correlation Matrix: correlation between every pair of numeric variables - the influence of the magnitude is removed

$$cor(x,y) = \frac{cov(x,y)}{\sigma_x \sigma_y}$$

Pearson Correlation Coefficient (ρ):

- measures the linear correlation between two variables;
- it has a value between +1 and -1.



Pearson Correlation Coefficient - cont.

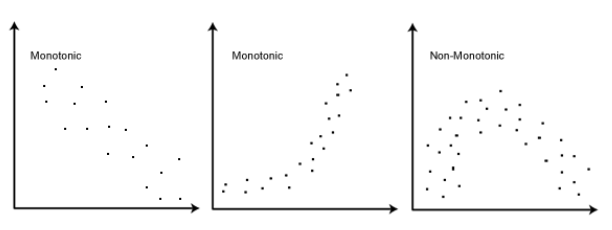
For a given sample of two variables x and y, $\{(x_1, y_1), ..., (x_n, y_n)\}$, the correlation coefficient is defined as

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

where n is the sample size, x_i and y_i are the individual sample points and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean, the same for $\bar{y}$

Spearman Rank-Order Correlation Coefficient:

- measures the strength and direction of monotonic association between two variables;
- two variables can be related according to a type of non-linear but still monotonic relationship.



Spearman Rank-Order Correlation Coefficient: cont.

- a rank-based, and non-parametric, version of *Pearson* correlation coefficient;
- it has a value between +1 and -1;

$$rs_{xy} = r_{rank_x rank_y}$$

- if all n ranks are distinct integers, it can be computed using the popular formula

$$rs_{xy} = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}$$

where $d_i = rank_{x_i} - rank_{y_i}$ is the difference between the two ranks of each observation.

Data Visualization

Data Visualization

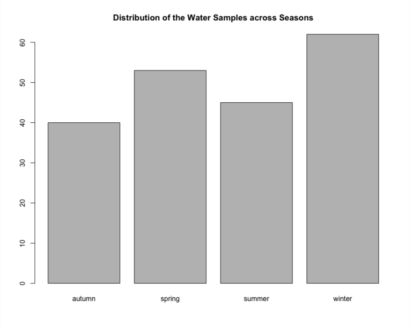
- Motivation
  - Humans are outstanding at detecting patterns and structures with their eyes
  - Data visualization methods try to explore these capabilities
  - Help detecting patterns and trends, and also outliers ans unusual patterns
- Main Types of Graphs
  - Univariate Graphs
  - Bivariate Graphs
  - Multivariate / Conditioned Graphs

Data Visualization: Univariate Graphs

- Categorical Variables
  - Barplots
  - Piecharts
  - ...
- Numeric Variables
  - Line plots
  - Histograms
  - QQ Plots
  - Boxplots
  - ...

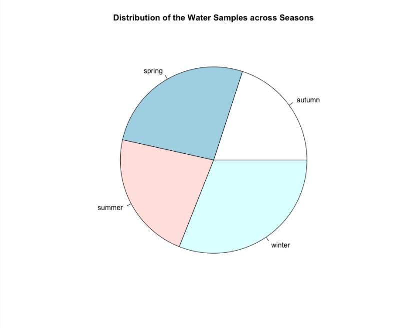
Barplots

- The main purpose is to display a set of values as heights of bars
- It can be used to display the frequency of occurrence of different values of a categorical variable



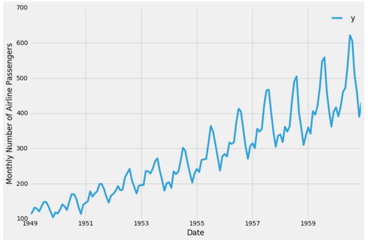
Piecharts

- Have the same purpose as bar plots but with information in the form of a pie.
- Are not so good for comparison purposes



Line Plots

- The main purpose is to to analyze the evolution of the values of a continuous variable.
- x-axis represent a quantitative scale with equal lag between observations.
- Frequently used to deal with the notion of time

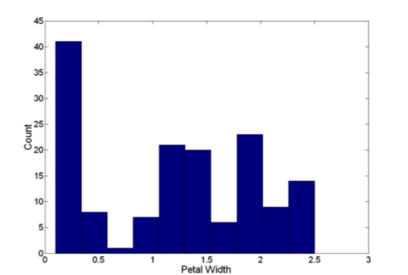


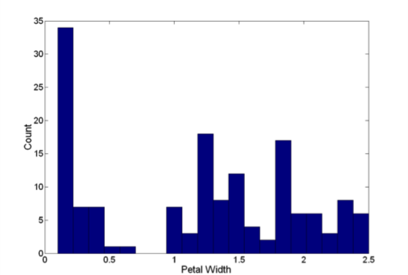
Histograms

- The main purpose is to display how the values of a continuous variable are distributed
- It is obtained as follows:
  - first, the range of the variable is divided into a set of bins (intervals of values)
  - then, the number of occurrences of values on each bin is counted
  - then, this number is displayed as a bar

Problems with Histograms

- Histograms may be misleading in small data sets
- The shape of the histogram depends on the number of bins
- How are the limits of the bins chosen? There are several algorithms for this.





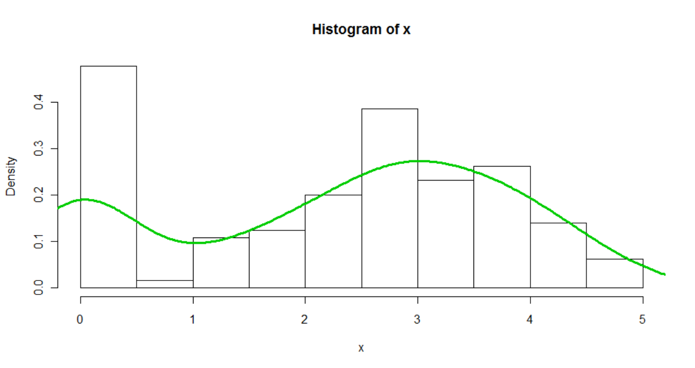
Data Visualization: Univariate Graphs (cont.)

- Some of the problems of histograms can be tackled by smoothing the estimates of the distribution of the values. That is the purpose of kernel density estimates
- Kernel estimates calculate the estimate of the distribution at a certain point by smoothly averaging over the neighboring points
- Namely, the density is estimated by

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

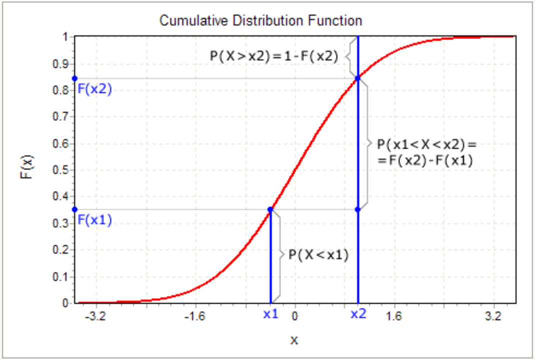
- where $K(.)$ is the kernel — a non-negative function — and $h > 0$ is a smoothing parameter called the bandwidth.

- Histogram with density estimate



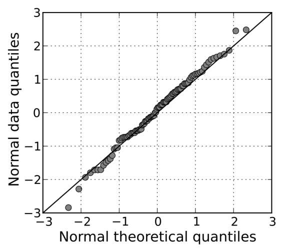
Cumulative Distribution Function (CDF)

- CDF of a random variable X: $F_X(x) = P(X \leq x)$

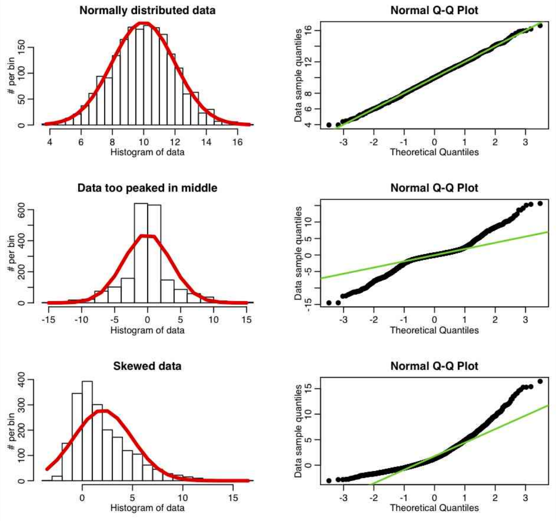


QQ Plots

- A graphical view of how properties such as location, scale, and skewness compare in two distributions.
- Can be used to visually check the hypothesis that the variable under study follows a normal distribution, comparing the observed distribution against the Normal distribution.

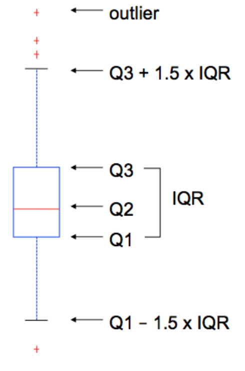


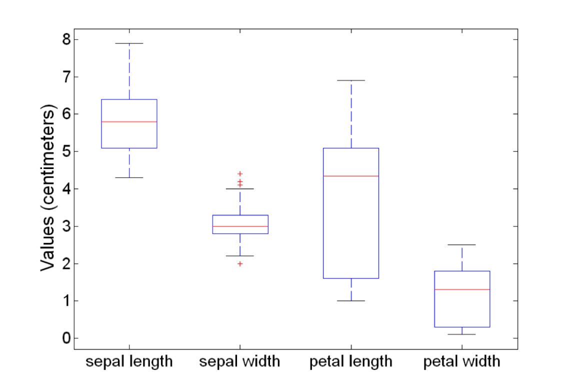
Data Visualization: Univariate Graphs (cont.)



Boxplots

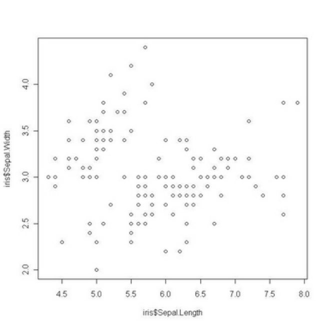
- Box plot provide an interesting summary of a variable distribution
- For instance, they inform us of the interquartile range and of the outliers (if any)





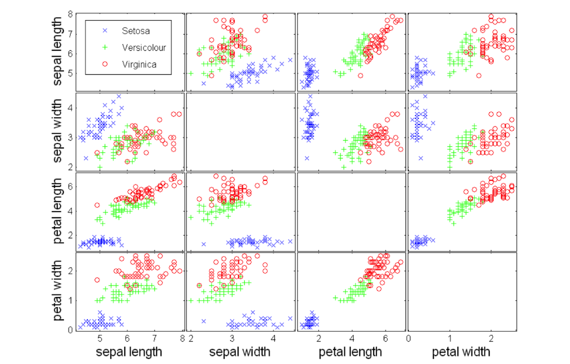
Scatterplots

- The natural graph for showing the relationship between two numeric variables



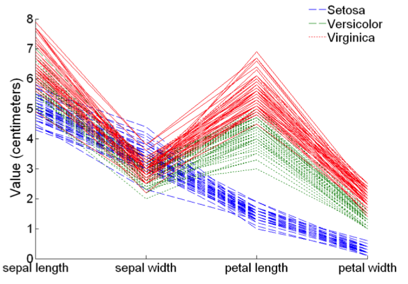
Data Visualization: Multivariate Graphs

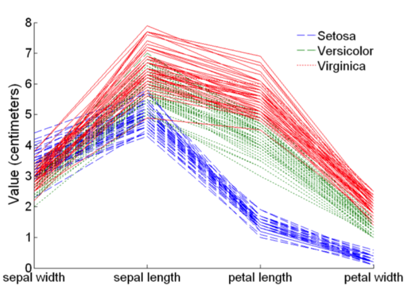
- The scatterplot can plot the relationship between every pair of numeric variables and respective groups



Parallel Coordinates Plot

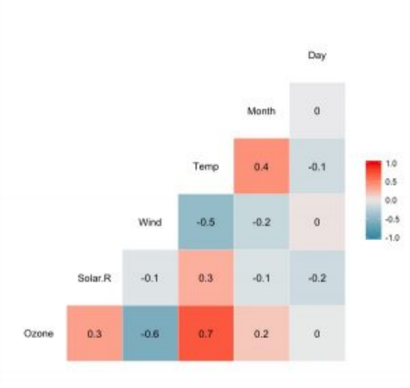
- Plots attributes values for each case (represented as a line)
- The order might be important to help identifying groups





Correlogram

- Chart of correlation statistics (e.g. pearson) for each pair of variables.



Conditioned Graphs

- Data sets frequently have categorical variables, which values can be used to create sub-groups of the data.
  - e.g. the sub-group of male clients of a company
- Conditioned plots allow the simultaneous presentation of these sub-group graphs to better allow finding eventual differences between the sub-groups
  - Conditioned Histograms
  - Conditioned Boxplots
  - . . .

The Grammar of Graphics in R

The Grammar of Graphics in R: `ggplot2`

- Package **`ggplot2`** implements the ideas created by Wilkinson (2005) on a grammar of graphics
- This grammar is the result of a theoretical study on what is a statistical graphic
- **`ggplot2`** builds upon this theory by implementing the concept of a layered grammar of graphics (Wickham, 2009)
- The grammar defines a statistical graphic as:
  - a mapping from data into aesthetic attributes (color, shape, size, etc.) of geometric objects (points, lines, bars, etc.)

The Grammar of Graphics in R: `ggplot2` (cont.)

- Main idea: specify the layers that make up the graphic, independently, and add them together with **+**
- Key elements of a statistical graphic:
  - data
  - aesthetic mappings
  - geometric objects
  - statistical transformations
  - scales
  - coordinate system
  - faceting
  - labelling

Aesthetic Mappings

- Controls the relation between data variables and graphic variables
  - e.g., map the Temperature variable of a data set into the *x*-axis in a scatter plot
  - e.g., map the Species of a plant into the *color* of dots in a graphic
- Some examples
  - position: **`aes(x=...,y=...)`**
  - color: **`aes(color=...)`**
  - fill: **`aes(fill=...)`**
  - shape: **`aes(shape=...)`**
  - linetype: **`aes(linetype=...)`**
  - size: **`aes(size=...)`**

The Grammar of Graphics in R: `ggplot2` (cont.)

Geometric Objects

- Controls what is shown in the graphics
  - e.g., show each observation by a point using the aesthetic mappings that relate two variables in the data set into the *x*-axis, *y*-axis of the graphic
- Some Examples:
  - scatterplot: **`geom_point()`**
  - line plot: **`geom_line()`**
  - boxplot: **`geom_boxplot()`**
  - histogram: **`geom_histogram()`**
  - barplot: **`geom_bar()`**

The Grammar of Graphics in R: `ggplot2` (cont.)

Statistical Tranformations

- Calculates and performs statistical analysis over the data in the graphic
  - e.g., count occurrences of certain values
  - e.g., discretize by creating bins
  - e.g., calculate the density by a density estimation function
- Some Examples:
  - **`stat_count(geom="bar")`** / **`geom_bar(stat="count")`**
  - **`stat_bin(geom="bar")`** / **`geom_histogram(stat="bin")`**
  - **`stat_density(geom="area")`** / **`geom_area(stat="density")`**
  - **`..count..,..density..`**: variables created by the statistic

Scales

- Maps the data values into values in the coordinate system of the graphics device

Scale	Types	Examples
`scale_color_`	identity	`scale_color_discrete()`
`scale_fill_`	manual	`scale_fill_continuous()`
`scale_size_`	continuous	`scale_size_manual()`
	discrete	`scale_size_discrete()`
`scale_shape_`	discrete	`scale_shape_discrete()`
`scale_linetype_`	identity	`scale_shape_manual()`
	manual	`scale_linetype_discrete()`
`scale_x_`	continuous	`scale_x_continuous()`
`scale_y_`	discrete	`scale_y_discrete()`
	reverse	`scale_x_reverse()`
	log	`scale_y_log()`
	date	`scale_x_date()`
	datetime	`scale_y_datetime()`

The Grammar of Graphics in R: `ggplot2` (cont.)

Coordinate System

- The coordinate system used to plot the data
- Some Examples:
  - Cartesian: **`coord_cartesian()`**
  - Polar: **`coord_polar()`**

Faceting

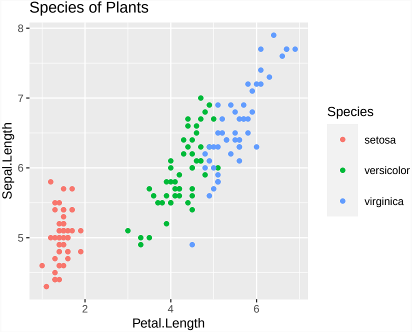
- Split the data into sub-groups and draw sub-graphs for each group (Conditioned Graphs)
- Examples:
    - **`facet_wrap()`**: defines groups according to the nominal values of a categorical variable
    - **`facet_grid()`**: defines groups according to the crossing of nominal values of two categorical variables

Labelling

- Label *x*-axis, *y*-axis, title of the graphic
- Some examples:
  - **`ggtitle()`**
  - **`xlab()`**
  - **`ylab()`**
  - **`labs(title=...,x=...,y=...)`**

Example 1: scatterplot

```
ggplot(iris, aes(x = Petal.Length, y = Sepal.Length,
    color = Species)) + geom_point() + ggtitle("Species of Plants")
```

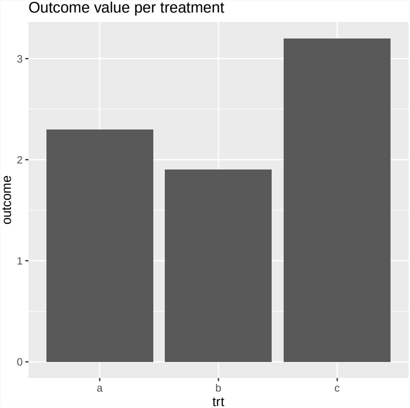


The Grammar of Graphics in R: `ggplot2` (cont.)

- There are two types of bar charts
  - **`geom_col()`**
    - makes the height of the bar representing values in the data
  - **`geom_bar()`**
    - makes the height of the bar proportional to the number of cases in each group

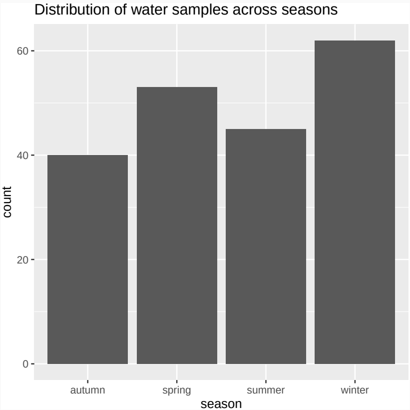
Example 2: barplot - I

```
ggplot(df, aes(x = trt, y = outcome)) + geom_col() +
    ggtitle("Outcome value per treatment")
```



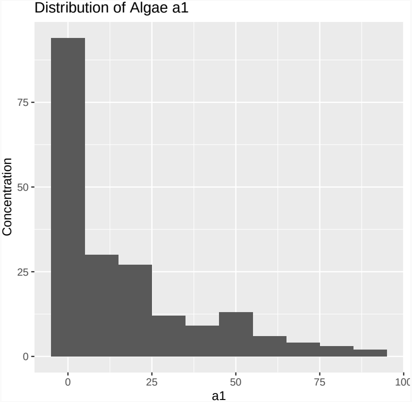
Example 3: barplot - II

```
ggplot(algae, aes(x = season)) + geom_bar() +
    ggtitle("Distribution of water samples across seasons")
```



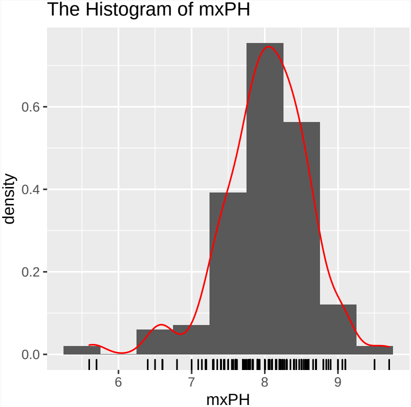
Example 4: histogram

```
ggplot(algae, aes(x = a1)) + geom_histogram(binwidth = 10) +
    ggtitle("Distribution of Algae a1") + ylab("Concentration")
```



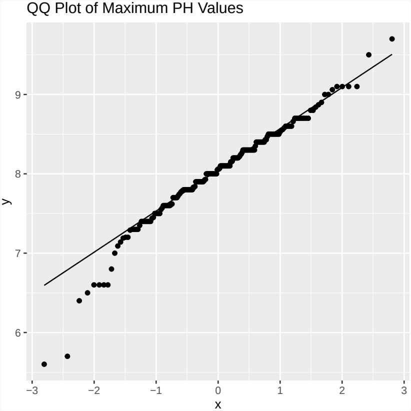
Example 5: histogram with density estimation

```
ggplot(algae, aes(x = mxPH)) + geom_histogram(binwidth = 0.5,
    aes(y = ..density..)) + geom_density(color = "red") + geom_rug() +
    ggtitle("The Histogram of mxPH")
```



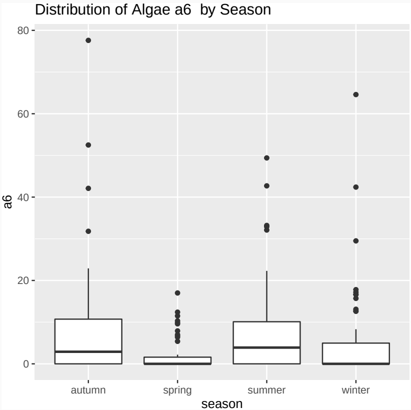
Example 6: QQ plot

```
ggplot(algae, aes(sample = mxPH)) + geom_qq(geom = "point") +
    stat_qq_line() + ggtitle("QQ Plot of Maximum PH Values")
```



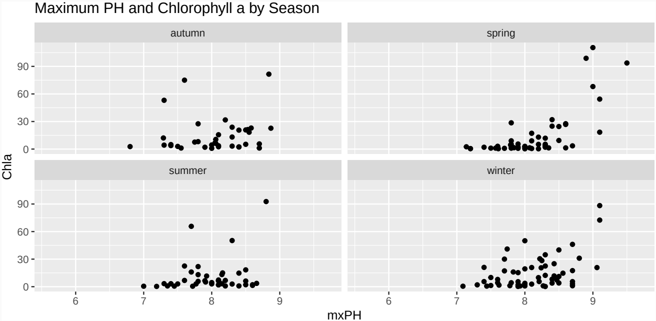
Example 7: conditioned boxplot

```
ggplot(algae, aes(x = season, y = a6)) +
    geom_boxplot() + ggtitle("Distribution of Algae a6  by Season")
```



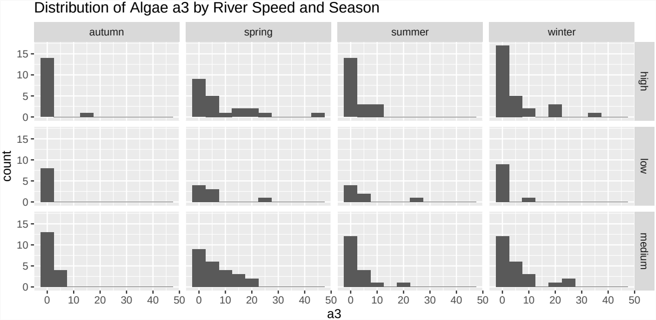
Example 8: conditioned scatterplot

```
ggplot(algae, aes(x = mxPH, y = Chla)) +
    geom_point() + facet_wrap(~season) +
    ggtitle("Maximum PH and Chlorophyll a by Season")
```



Example 9: conditioned histogram

```
ggplot(algae, aes(x = a3)) + geom_histogram(binwidth = 5) +
    facet_grid(speed ~ season) + ggtitle("Distribution of Algae a3 by River Spee
```



The Grammar of Graphics in R: `ggplot2` (cont.)

- Many more insteresting things can be done with `ggplot2`
- For a more complete reference
  - R Graphics Cookbook, 2nd edition

Data Preparation

Data Preparation

Set of steps that may be necessary to carry out before any further analysis takes place on the available data.

- Data can come from a multitude of different sources
- Frequently, we have data sets with unknown variable values
- Many data mining methods are sensitive to the scale and/or the type of variables
    - Different variables may have different scales
    - Some methods are unable to handle either nominal or numerical variables

Data Preparation (cont.)

- We may face the need to “create” new variables to achieve our objectives
  - Sometimes we are more interested in relative values (variations) than absolute values
  - We may be aware of some domain-specific mathematical relationship among two or more variables that is important for the task
- Our data set may be too large for some methods to be applicable

Data Preparation (cont.)

- Feature Extraction
  - extract features from raw data on which analysis can be performed.
- Data Cleaning
  - data may be hard to read or require extra parsing efforts.
- Data Transformation
  - it may be necessary to change some of the values of the data.
- Feature Engineering
  - to incorporate some domain knowledge.
- Data and Dimensionality Reduction
  - to make modeling possible.

Feature Extraction

- It is very application specific and a very crucial step.
  - **sensor data**: large volume of low-level signals associated with date/time attributes
  - **image data**: very high-dimensional data that cane be represented by pixels, color histograms, etc.
  - **web logs**: text in a prespecified format with both categorical and numerical attributes
  - **network traffic**: network packets information
  - **document data**: raw and unstructured data

Ultimate Goal

- Making our data set tidy
    - each value belongs to a variable and an observation
    - each variable contains all values of a certain property measured across all observations
    - each observation contains all values of the variables measured for the respective case
- These properties lead to data tables where:
    - each row represents an observation
    - each column represents an attribute measured for each observation

Data Cleaning: Handling Missing Values (cont.)

Main Strategies

- Remove all cases in a data set with some unknown value
- Fill-in the unknowns with the imputation of the most common value (a statistic of centrality)
- Fill-in with the most common value on the cases that are more “similar” to the one with unknowns.
- Fill-in with linear interpolation of nearby values in time and/or space.
- Explore eventual correlations between variables
- Do nothing: many data mining methods are designated to work robustly with missing values

Data Cleaning: Handling Incorrect Values

- Inconsistency detection
  - data integration techniques within the database field
- Domain knowledge
  - data auditing that use domain knowledhe and constraints
- Data-centric methods
  - statistical-based methods to detect outliers

Data Transformation

- Map entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values
- Why it may be useful?
  - Imagine two attributes (e.g. age and salary) with a very different scale
  - Any aggregation function (e.g. euclidean distance) computed on the set of cases, will be dominated by the attribute of larger magnitude.
- Some common strategies:
  - Normalization
  - Binarization / One-Hot Enconding
  - Discretization

Data Transformation: Normalization

- Min-Max Scaling (Range-based Normalization)

$$y_i = \frac{x_i - \min_x}{\max_x - \min_x}$$

  - $\min_x$ and $\max_x$ are the minimum and maximum values of attribute x
  - values will lie in the range $[0, 1]$
- It is not robust for scenarios where there are outliers
  - if an erroneous age value of 800 is registered instead of 80, most of the values will be in the range $[0, 0.1]$

Data Transformation: Normalization (cont.)

- Standarization (z-score Normalization):

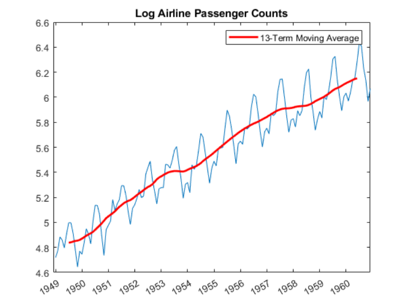
$$y_i = \frac{x_i - \mu_x}{\sigma_x}$$

  - μ_x and σ_x are the mean and the standard deviation of attribute x
  - values are rescaled so that they have $\mu_x = 0$ and $\sigma_x = 1$
  - values will, typically, lie in the range $[-3, 3]$ under a normal distribution assumption

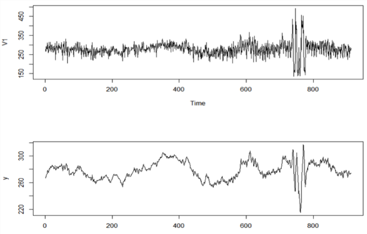
Data Transformation: Case Dependencies

- In time series it is common to use different techniques.
- Examples:
  - to adjust mean, variance, range
  - to remove unwnated, common signal

Moving Average



Low-pass filter



Data Transformation: Binarization / One-Hot Enconding

- Some data mining methods are only able to handle numeric attributes.
- If the categorical attribute is not ordinal, it is necessary to convert it into a numerical attribute.
- Binarization: if the atribute has only 2 possible nominal values, it can be transformed into 1 binary attribute
  - fever: yes/no $->$ fever: 1/0
- One-Hot Enconding: if the atribute has *k* possible nominal values, it can be transformed into *k* binary attributes
  - eye_color: brown/blue/green $\rightarrow$ eye_brown: 1/0, eye_blue: 1/0, eye_green: 1/0

Data Transformation: Discretization

- Process of converting a continuous attribute into an ordinal attribute of numeric variables.
- Some unsupervised discretization: find breaks in the data values
  - Equal-width
    - it divides the original values into equal-width range of values
    - it may be affected by the presence of outliers
  - Equal-frequency
    - it divides the original values so that the same number of values are assigned to each range
    - it can generate ranges with very different amplitudes
- Supervised discretization: use class labels to find breaks (we'll see later)

Feature Engineering

Fundamental to the application of machine learning.

'(...) some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used.' - Pedro Domingos, 2012

- The process of using domain knowledge of the data to create features that might help when solving the problem.
- New features that can capture the important information in a data set much more efficiently than the original features.
- Case 1: express known relationships between existing variables
  - create ratios and proportions like credit card sales per person
  - from web logs obtain the average session duration per user, the frequency of access, etc.

Feature Engineering: Cases Dependencies

- Case 2: overcome limitations of some data mining tools regarding cases dependencies.
  - some tools shuffle the cases, or are not able to use the information about their dependencies (time, space, space-time)
  - two main ways of handling this issue:
    - constrain ourselves to tools that handle these dependencies directly
    - create variables that express the dependency relationships
- In time series is common to create features that represent relative values instead of absolute values, so to avoid trend effects.

$$y_t = \frac{x_t - x_{t-1}}{x_{t-1}}$$

Feature Engineering: Cases Dependencies (cont.)

- Other common technique is Time Delay Embedding

- Create variables whose values are the value of the same variable in previous time steps

X_{t-3}	X_{t-2}	X_{t-1}	X_t
x_{t_1}	x_{t_2}	x_{t_3}	x_{t_4}
x_{t_2}	x_{t_3}	x_{t_4}	x_{t_5}
...			
$x_{t_{n-3}}$	$x_{t_{n-3}}$	$x_{t_{n-1}}$	x_{t_n}

- If we have variables whose values are the value of the same variable but on different time steps, standard tools will be able to model the time relationships with these embeddings

- Note that similar “tricks” can be done with space and space-time dependencies

References

References

Aggarwal, Charu C. 2015. *Data Mining, the Texbook*. Springer.

Gama, João, André Carlos Ponce de Leon Ferreira de Carvalho, Katti Faceli, Ana Carolina Lorena, and Márcia Oliveira. 2015. *Extração de Conhecimento de Dados: Data Mining -3rd Edition*. Edições Sílabo.

Gandomi, Amir, and Murtaza Haider. 2015. "Beyond the Hype: Big Data Concepts, Methods, and Analytics." *International Journal of Information Management* 35 (2): 137–44. doi:https://doi.org/10.1016/j.ijinfomgt.2014.10.007.

Han, Jiawei, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Moreira, João, Andre Carvalho, and Tomás Horvath. 2018. *Data Analytics: A General Introduction*. Wiley.

"R Project." 2021. https://www.r-project.org/.

Tan, Pang-Ning, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2018. *Introduction to Data Mining*. 2nd ed. Pearson.